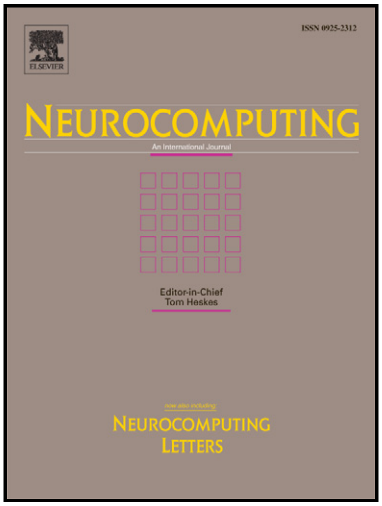

www.elsevier.com/locate/neucom

Discriminative structure selection method of gaussian mixture models with its application to handwritten digit recognition

Xuefeng Chen, Xiabi Liu, Yunde Jia

PII: S0925-2312(10)00471-6
DOI: doi:10.1016/j.neucom.2010.11.010
Reference: NEUCOM 12008

To appear in: *Neurocomputing*

Received date: 20 January 2010
Revised date: 4 November 2010
Accepted date: 8 November 2010

Discriminative Structure Selection Method of Gaussian Mixture Models with its Application to Handwritten Digit Recognition

Xuefeng Chen, Xiabi Liu*, Yunde Jia

Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

Abstract

Model structure selection is currently an open problem in modeling data via Gaussian Mixture Models (GMM). This paper proposes a discriminative method to select GMM structures for pattern classification. We introduce a GMM structure selection criterion based on a discriminative objective function called Soft target based Max-Min posterior Pseudo-probabilities (Soft-MMP). The structure and the parameters of the optimal GMM are estimated simultaneously by seeking the maximum value of Laplace's approximation of the integrated Soft-MMP function. The line search algorithm is employed to solve this optimization problem. We evaluate the proposed GMM structure selection method through the experiments of handwritten digit recognition on the well-known CENPARMI and MNIST digit databases. Our method behaves better than the manual method and the generative counterparts, including Bayesian Information Criterion (BIC), Minimum Description Length (MDL) and AutoClass. Furthermore, to our best knowledge, the digit classifier trained by using our method achieves the best error rate so far on the CENPARMI database and the error rate comparable to the currently best ones on the MNIST database.

Keywords**:** Gaussian Mixture Models (GMM), Structure selection, parameter estimation, discriminative learning, Finite Mixture Models (FMM), Max-Min posterior Pseudo-probabilities (MMP).

* Corresponding author. Tel.: +86 10 68913447, Fax: +86 10 86343158.
E-mail addresses: crocodel@bit.edu.cn (X. Chen), liuxiabi@bit.edu.cn (X. Liu), jiayunde@bit.edu.cn (Y. Jia).

1. Introduction

Gaussian Mixture Model (GMM) is a widely used statistical tool in pattern classification. They are flexible enough to approximate any given density with high accuracy [1]. In fitting GMM to data, we need to select the number of GMM components and estimate parameters in the GMM with certain number of components. The two tasks are usually known as structure selection and parameter estimation, respectively. Currently, the satisfactory results of GMM parameter estimation have been reported in many literatures. But how to select appropriate GMM structures is still a challenge.

Existing methods of GMM structure selection can be divided into five categories: cross-validation [2-3], stochastic methods [4], information theory approaches [5-7], infinite Gaussian Mixture Models [8-9], and Bayesian methods [1, 10-13]. The main idea of the last category is to evaluate model structures using the integral over parameters. Various criteria under Bayesian framework have been developed for GMM structure selection, including Bayesian Information Criterion (BIC) [10], Laplace criterion [11], variational Bayesian criterion [12], Laplace-Empirical criterion [13], etc. These works show that Bayesian methods are promising to solve the problem of model structure selection. However, most existing Bayesian methods are based on generative learning, usually on classical Maximum Likelihood Estimation (MLE), where only positive examples are involved to determine model structures. Therefore, the discriminative ability of the model is somewhat ignored. In recent years, discriminative learning algorithms such as Minimum Classification Error (MCE) [14], Maximum Mutual Information (MMI) [15], Minimum Phone Error (MPE) [16] and Max-Min Posterior Pseudo-Probabilities [17] have demonstrated their advantages over generative learning counterparts for parameter estimation of GMMs. Compared to the advances in discriminative parameter estimation, discriminative structure selection has not received enough attention. Recently, Klautau et al. [18] presented a MMI based method to determine the GMM structure. Liu and Gales [19] introduced a discriminative method of GMM complexity control under Bayesian model structure selection framework. In the method of Liu and Gales, a marginalized discriminative growth function of MMI\MPE criterion was presented to select the GMM structure. They evaluated their method in large-vocabulary continuous-speech recognition.

In this paper, we propose a discriminative GMM structure selection method for pattern classification through embedding a discriminative learning criterion into Bayesian model structure selection framework. The used discriminative learning criterion is SOFT target based Max-Min posterior Pseudo-probabilities (Soft-MMP) [20]. An integrated Soft-MMP function is introduced and approximated with Laplace's method, the value of which is used to evaluate the GMM structure. By employing the line search algorithm to find out the maximum value of Laplace's approximation of the integrated Soft-MMP function, the structure and the parameters of the optimal GMM are determined simultaneously in a discriminative manner. Our work is closely related to that of Liu and

Gales [19], but the model evaluation criteria are different from each other. We advise a Soft-MMP based criterion in this paper, while Liu and Gales designed MMI or MPE based one. Furthermore, we employ the line search algorithm for model structure selection, while the merge-split strategy is used by Liu and Gales.

Our method was applied to handwritten digit recognition and evaluated on two well-known handwritten digit databases, CENPARMI [21] and MNIST [22]. We compare our method with the manual method and three main generative counterparts, including BIC [10], Minimum Description Length (MDL) [7], and AutoClass [23]. The comparison results show that the proposed method improves both the recognition rates and the generalization ability of GMM based handwritten digit classifiers. Compared with these GMM structure selection methods, our method brings (1) 27.78% to 51.85% reduction in the error rate on the CENPARMI test set and 15.87% to 33.75% reduction for the MNIST test set; (2) 0.18% to 0.45% increase in the generalization ability which is measured as the ratio of the recognition rate on the test set to that on the training set for the CENPARMI database and 0.06% to 0.17% increase for the MNIST database.

Furthermore, to our best knowledge, our method brings the best error rate so far on the CENPARMI database and the error rate comparable to the currently best ones on the MNIST database. In the work of Liu et al. [24-25], the state-of-the-art techniques of handwritten digit recognition, including features and classifiers, is thoroughly investigated on both CENPARMI and MNIST database. They use 8-direction gradient features (abbreviated to e-grg there) and the classifier of either SVM with RBF kernel or Discriminative Learning Quadratic Discriminant Function (DLQDF) to report the error rate of 0.95% on the test set of the CENPARMI database. Using the same e-grg features by courtesy of Liu, we achieve the better error rate of 0.65% on the same test set. This result also outperforms the other up-to-date results reported on the CENPARMI database by using various features and classifiers [24-29], including previous best result of 0.85% from SVM with RBF kernel and deslant chaincode feature [25]. For the MNIST database, our method achieves the error rate of 0.53% on the test set by using e-grg feature. This result is comparable to the best error rate of 0.42% for e-grg feature [24] and the overall best error rate of 0.39% [30] on the same database.

The rest of this paper is organized as follows. Section 2 describes Bayesian model structure selection framework and Soft-MMP discriminative learning criterion. Section 3 presents our discriminative method of GMM structure selection. Section 4 reports the experimental evaluation of our method for handwritten digit recognition. We discuss our conclusions and future work in Section 5.

2. Preliminaries

In this section, we briefly introduce Bayesian structure selection and Soft-MMP. The reader is referred to Liu and Gales [19] for more details of Bayesian model structure selection and Chen [20] for more details of Soft-MMP.

2.1. Bayesian model structure selection

Let M and $\boldsymbol{\Lambda}$ be the number of components and the set of unknown parameters of a GMM, $p(\boldsymbol{\Lambda}|M)$ be the parameter prior distribution, $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ be a training data set of N examples. Then the integrated likelihood for the model is

$$p(\boldsymbol{X}|M) = \int p(\boldsymbol{X}|\boldsymbol{\Lambda}, M)p(\boldsymbol{\Lambda}|M)d\boldsymbol{\Lambda}. \tag{1}$$

In Bayesian model structure selection methods, the optimal model structure will be determined by maximizing the integrated likelihood:

$$M^* = \arg\max_M \int p(\boldsymbol{X}|\boldsymbol{\Lambda}, M)p(\boldsymbol{\Lambda}|M)d\boldsymbol{\Lambda}. \tag{2}$$

Following [19], $p(\boldsymbol{\Lambda}|M)$ is treated as uninformative. Therefore, the optimal model structure is computed by

$$M^* = \arg\max_M \int p(\boldsymbol{X}|\boldsymbol{\Lambda}, M)d\boldsymbol{\Lambda}. \tag{3}$$

The integrated likelihood in Eq. (3) is usually a high-dimensional and intractable integral. Various analytic and numerical approximations have been proposed. We use Laplace's approximation [10] in this paper, which can be expressed as

$$p(\boldsymbol{X}|M) \approx p(\boldsymbol{X}|\boldsymbol{\Lambda}^*, M)\sqrt{\frac{(2\pi)^S}{\left|-\nabla^2_{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^*} \log p(\boldsymbol{X}|\boldsymbol{\Lambda}, M)\right|}}, \tag{4}$$

where $\boldsymbol{\Lambda}^*$ is the MLE of parameters, S is the number of parameters, and $|\cdot|$ denotes the determinant of a matrix.

2.2. Soft-MMP

The Soft-MMP is developed to estimate parameters in the posterior pseudo-probability based classifier [17], a recently proposed variant of Bayesian classifier. Let $\boldsymbol{x}$ be a feature vector, C_i be the i-th class, and $p(\boldsymbol{x}|C_i)$ be the class-conditional probability density function. Then the posterior pseudo-probability of being C_i for $\boldsymbol{x}$ is computed by

$$f(p(\boldsymbol{x}|C_i)) = 1 - \exp(-\kappa p^{\beta}(\boldsymbol{x}|C_i)), \tag{5}$$

where κ and β are positive numbers. For any input pattern, we compute the corresponding posterior pseudo-probabilities of all the classes under consideration. Then the input pattern is classified as the class C^* with the maximum posterior pseudo-probability:

$$C^* = \arg\max_{C_i} f\left(p\left(\boldsymbol{x}|C_i\right)\right). \tag{6}$$

According to Eq. (6), the posterior pseudo-probability is in direct proportional to the class-conditional probability density, so the classification decision made by posterior pseudo-probabilities is consistent with that by traditional Bayesian counterpart which assumes the prior probabilities of all the classes are equal. However, posterior pseudo-probabilities take values in $[0,1]$, by introducing which discriminative learning approaches such as MMP [17] and Soft-MMP [20] can be developed for Bayesian classifiers. Furthermore, the posterior pseudo-probability is a natural similarity measure and is useful for (1) making rejection decision, (2) combining classifiers, and (3) assessing the performance of a classifier in a much more accurate way than that of counting the number of patterns classified correctly [31].

The class-conditional probability density function in Eq. (6) should be provided for constructing a posterior pseudo-probabilities based classifier, which is assumed to be the GMM in this paper. Let M be the number of GMM components; w_k, $\boldsymbol{\mu_k}$ and $\boldsymbol{\Sigma_k}$ be the weight, the mean, and the covariance matrix of the k-th Gaussian component, respectively. $\sum_{k=1}^{M} w_k = 1$. Then we have

$$p\left(\boldsymbol{x}|C\right) = \sum_{k=1}^{M} w_k N_k\left(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right), \tag{7}$$

where

$$N_k\left(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) = (2\pi)^{-\frac{d}{2}} \left|\boldsymbol{\Sigma}_k\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}_k\right)^T \boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}_k\right)\right). \tag{8}$$

By substituting Eq. (7) into Eq. (6), we get a posterior pseudo-probabilities based classifier. The original Soft-MMP learning method is able to estimate the parameters in this classifier, but the GMM structure needs to be set manually.

Let $\hat{H}$ and $\bar{H}$ be two adaptive soft targets which take values in $[0,1]$; $\hat{\boldsymbol{x}}$ and $\bar{\boldsymbol{x}}$ be the feature vector of arbitrary positive and negative example of a class, respectively; m and n be the number of positive and negative examples of the class in the training set, respectively. Then the total empirical loss of the posterior pseudo-probabilities based classifier is measured to be

$$L(\boldsymbol{\Lambda}, M) = \frac{1}{m}\sum_{i=1}^{m} \hat{l}^2\left(\hat{\boldsymbol{x}}_i; \boldsymbol{\Lambda}, M\right) + \frac{1}{n}\sum_{i=1}^{n} \bar{l}^2\left(\bar{\boldsymbol{x}}_i; \boldsymbol{\Lambda}, M\right), \tag{9}$$

where $\boldsymbol{\Lambda}$ is the parameter set, $\hat{l}\left(\hat{\boldsymbol{x}}; \boldsymbol{\Lambda}, M\right)$ is the empirical lose of the classifier on positive examples:

$$\hat{l}(\hat{\boldsymbol{x}};\boldsymbol{\Lambda},M)=\begin{cases}0 & , \quad f(p(\hat{\boldsymbol{x}}|C_i))>\hat{H} \\ \hat{H}-f(p(\hat{\boldsymbol{x}}|C)), & \quad f(p(\hat{\boldsymbol{x}}|C_i))\le\hat{H}\end{cases}, \tag{10}$$

and $\bar{l}(\bar{\boldsymbol{x}};\boldsymbol{\Lambda},M)$ is the empirical lose of the classifier on negative examples:

$$\bar{l}(\bar{\boldsymbol{x}};\boldsymbol{\Lambda},M)=\begin{cases}0 & , \quad f(p(\bar{\boldsymbol{x}}|C_i))<\overline{H} \\ f(p(\bar{\boldsymbol{x}}|C_i))-\overline{H}, & \quad f(p(\bar{\boldsymbol{x}}|C_i))\ge\overline{H}\end{cases}. \tag{11}$$

The objective of Soft-MMP is to minimize the empirical loss and maximize the difference between $\hat{H}$ and $\overline{H}$, which can be formally described as

$$F(\boldsymbol{\Lambda},M)=\omega(1-d)^2+(1-\omega)L(\boldsymbol{\Lambda},M). \tag{12}$$

In Eq. (12), $d=\hat{H}-\overline{H}$, and ω is a non-negative constant to control the tradeoff between the empirical loss and the difference between two soft targets.

Consequently, the task of Soft-MMP learning is to find out the optimal parameter set $\boldsymbol{\Lambda}^*$ by minimizing $F(\boldsymbol{\Lambda},M)$:

$$\boldsymbol{\Lambda}^*=\arg\min_{\Lambda}F(\boldsymbol{\Lambda},M). \tag{13}$$

In the next section, this minimization problem is transformed to the maximization one for defining our model structure selection criterion.

3. The proposed method

In this section, we present our discriminative method of GMM structure selection based on Laplace's approximation of the integrated Soft-MMP function. We firstly describe our model structure selection criterion and its Laplace approximation. We then give a line search algorithm for finding out the optimal GMM structure and parameters.

3.1. Inverse Soft-MMP function

Our evaluation criterion of GMM is defined by replacing the likelihood function in the Bayesian evidence integral (1) with the discriminative Soft-MMP function. However, the original Soft-MMP learning is a minimization problem. The integrated Soft-MMP function cannot be approximated with Laplace's method. In order to remove this obstacle, the original Soft-MMP function (12) is rewritten as

$$\widetilde{F}(\boldsymbol{\Lambda},M)=\lambda\left(1.0-\overline{H}+\hat{H}\right)+\left(mn-\frac{n}{2}\sum_{i=1}^{m}\hat{l}^2(\hat{\boldsymbol{x}}_i;\boldsymbol{\Lambda})-\frac{m}{2}\sum_{j=1}^{n}\bar{l}^2(\bar{\boldsymbol{x}}_i;\boldsymbol{\Lambda})\right). \tag{14}$$

The first term in the right part of Eq. (14) stands for the distance between two soft targets and the second term for the empirical loss. These two terms are balanced by the tradeoff parameter λ. We can obtain the optimal parameter set $\boldsymbol{\Lambda}^*$ by maximizing $\widetilde{F}(\boldsymbol{\Lambda},M)$:

$$\boldsymbol{\Lambda}^*=\arg\max_{\Lambda}\widetilde{F}(\boldsymbol{\Lambda},M). \tag{15}$$

The objective expressed in Eq. (15) is the same as the original one in Eq. (13). Both of them try to minimize the empirical loss and maximize the difference between two soft targets. However, the Soft-MMP learning based on Eq. (15) is a maximization problem. It is feasible to compute Laplace's approximation of the corresponding integrated function.

3.2. Discriminative criterion for GMM structure selection

According to Eq. (15), we can determine the optimal parameter set $\boldsymbol{\Lambda}^*$ for specific GMM structure. Then the optimal GMM structure M^* will be selected as

$$M^* = \arg\max_{M}\left\{2\pi\widetilde{F}(\boldsymbol{\Lambda}^*, M) - \frac{1}{2}\sum_{i=1}^{S}\log\left(-\nabla^2_{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^*}\widetilde{F}(\boldsymbol{\Lambda}_i, M)\right)\right\}, \tag{16}$$

where $\nabla^2\widetilde{F}(\boldsymbol{\Lambda}_i, M)\Big|_{i=1}^{S}$ is the second order partial derivatives of $\widetilde{F}(\boldsymbol{\Lambda}, M)$ with respect to each parameter in $\boldsymbol{\Lambda}$. This selection criterion of GMM structure is obtained by substituting $\widetilde{F}(\boldsymbol{\Lambda}, M)$ for $p(\boldsymbol{X}|\boldsymbol{\Lambda}, M)$ in Eq. (3) and then performing computational simplifications. The detailed derivation is given as follows.

Firstly, replacing $p(\boldsymbol{X}|\boldsymbol{\Lambda}, M)$ in Eq. (3) with $\widetilde{F}(\boldsymbol{\Lambda}, M)$, we get

$$M^* = \arg\max_{M} \int\widetilde{F}(\boldsymbol{\Lambda}, M)d\boldsymbol{\Lambda}. \tag{17}$$

Laplace's approximation of $\int\widetilde{F}(\boldsymbol{\Lambda}, M)d\boldsymbol{\Lambda}$ is

$$\int\widetilde{F}(\boldsymbol{\Lambda}, M)d\boldsymbol{\Lambda} \approx \left(\widetilde{F}(\boldsymbol{\Lambda}, M)\sqrt{\frac{(2\pi)^S}{\left|-\nabla^2_{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^*}\log\widetilde{F}(\boldsymbol{\Lambda}, M)\right|}}\right). \tag{18}$$

Since the arithmetic overflow is possible to occur when computing $\int\widetilde{F}(\boldsymbol{\Lambda}, M)d\boldsymbol{\Lambda}$, $\log\int\widetilde{F}(\boldsymbol{\Lambda}, M)d\boldsymbol{\Lambda}$ is considered here. We further introduce

$$\widehat{F}(\boldsymbol{\Lambda}, M) = \exp\left(2\pi\widetilde{F}(\boldsymbol{\Lambda}, M)\right), \tag{19}$$

which is directly proportional to $\widetilde{F}(\boldsymbol{\Lambda}, M)$. Laplace's approximation of $\log\int\widehat{F}(\boldsymbol{\Lambda}, M)d\boldsymbol{\Lambda}$ is

$$\log\int\widehat{F}(\boldsymbol{\Lambda}, M)d\boldsymbol{\Lambda} \approx \log\left(\widehat{F}(\boldsymbol{\Lambda}^*, M)\sqrt{\frac{(2\pi)^S}{\left|-\nabla^2_{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^*}\log\widehat{F}(\boldsymbol{\Lambda}, M)\right|}}\right)$$

$$= 2\pi\widetilde{F}(\boldsymbol{\Lambda}^*, M) + \frac{1}{2}\log\left(\frac{(2\pi)^S}{\left|-\nabla^2_{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^*}2\pi\widetilde{F}(\boldsymbol{\Lambda}^*, M)\right|}\right). \tag{20}$$

As shown in Eq. (19) and (20), we can get the same structure selection result based on $\log \int \widetilde{F}(\boldsymbol{\Lambda}, M) d\boldsymbol{\Lambda}$ or $\log \int \widehat{F}(\boldsymbol{\Lambda}, M) d\boldsymbol{\Lambda}$, but the computation is simplified by introducing $\widehat{F}(\boldsymbol{\Lambda}, M)$.

A complex model often contains many parameters, such as more than 2000 in our experiments. So the cost of calculating the Hessian matrix $\nabla^2 \widetilde{F}(\boldsymbol{\Lambda}, M)$ is very expensive. In this paper, the Hessian matrix is assumed to have a diagonal structure for making the problem tractable. The assumption works well in our experiments. Thus we have

$$
\begin{aligned}
\log \int \widehat{F}(\boldsymbol{\Lambda}, M) d\boldsymbol{\Lambda} &= 2\pi \widetilde{F}(\boldsymbol{\Lambda}^*, M) + \frac{1}{2} \log\left(\frac{(2\pi)^S}{\left| -\nabla^2_{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^*} 2\pi \widetilde{F}(\boldsymbol{\Lambda}^*, M) \right|} \right) \\
&= 2\pi \widetilde{F}(\boldsymbol{\Lambda}^*, M) + \frac{1}{2} \log\left(\frac{(2\pi)^S}{(2\pi)^S \left| -\nabla^2_{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^*} \widetilde{F}(\boldsymbol{\Lambda}^*, M) \right|} \right) \\
&= 2\pi \widetilde{F}(\boldsymbol{\Lambda}^*, M) - \frac{1}{2} \log\left(\left| -\nabla^2_{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^*} \widetilde{F}(\boldsymbol{\Lambda}^*, M) \right| \right) \\
&= 2\pi \widetilde{F}(\boldsymbol{\Lambda}^*, M) - \frac{1}{2} \sum_{i=1}^{S} \log\left(-\nabla^2_{\boldsymbol{\Lambda}_i=\boldsymbol{\Lambda}_i^*} \widetilde{F}(\boldsymbol{\Lambda}_i, M) \right). \qquad (21)
\end{aligned}
$$

Finally, we get our GMM structure selection criterion:

$$
\begin{aligned}
M^* &= \arg\max_M \log \int \widehat{F}(\boldsymbol{\Lambda}, M) d\boldsymbol{\Lambda} \\
&= \arg\max_M \left\{ 2\pi \widetilde{F}(\boldsymbol{\Lambda}^*, M) - \frac{1}{2} \sum_{i=1}^{S} \log\left(-\nabla^2_{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^*} \widetilde{F}(\boldsymbol{\Lambda}_i, M) \right) \right\}. \qquad (22)
\end{aligned}
$$

3.3. Optimization algorithm

3.3.1. Structure selection

In most of existing model structure selection methods, the exhaustive search strategy is used and the search interval is decided manually. Since Laplace's method approximates the integral of a function by computing the volume under a Gaussian, Laplace's approximation of the integrated Soft-MMP function, i.e. Eq. (21), is a unimodal function with respect to M. So we employ the line search algorithm to seek the maximum value of Eq. (21), which includes two stages.

In the first stage, an initial search interval of the number of GMM components is determined by the advance-retreat method [32]. The algorithm starts from a triggering number to find three numbers in a monotonic direction. The values of Eq. (21) corresponding to these three numbers should show "low-high-low" trend. If the algorithm fails to find numbers satisfying the condition, it retreats to the triggering number and performs the similar search in the opposite direction. In the second stage, the search interval is reduced continuously by the golden section method [33] until a maximum value of Eq. (22) is reached. The golden section method compares function values on two tentative numbers in and two end-numbers of search interval. Suppose the four numbers are arranged in ascending order from left to right. If the maximum function value comes from one of the left-hand two points, the search interval is reduced by discarding the right end-number. Otherwise, the search interval is reduced by discarding the left end-number. This procedure is iteratively performed until none tentative points can be selected in the search interval. The number corresponding to the maximum value of Eq. (22) in the final search interval is outputted as the optimal result. It should be noted that the discrete numbers are searched in the process above. Two tentative value calculated in each iteration of the golden section method will be rounded down (lower value) or up (higher value). The details of the two stages in the line search algorithm are given in Table 2.

The computational complexity of the line search algorithm and the exhaustive search algorithm are analyzed and compared with each other in the following. Suppose the length of the initial search interval is L. By using the golden section method, the search interval will be reduced to $(0.618)^n L$ after n iterations. Since the discrete space is explored, the searching must terminate if $(0.618)^n L \leq 1$. So the upper bound of iteration time for the golden section method is

$$n \leq 1 - \log_{0.618}^{L} . \tag{23}$$

As for the exhaustive search algorithm, the computational complexity is usually measured by the mean acquisition time, $E(L)$, which is the expected value of finding each point within the search interval. Since $n+1$ iteration time is required by the exhaustive search strategy to find the n-th point, we have

$$E(L) = \frac{(2+3+\cdots+L+1)}{L} = 1 + \frac{L+1}{2} . \tag{24}$$

Comparing Eq. (23) to Eq. (24), we can conclude that the efficiency of the golden section method is better than the exhaustive search strategy if the search interval is large enough. Fig. 1 shows that the advantage of the golden section method becomes more and more obvious when the search interval is increased.

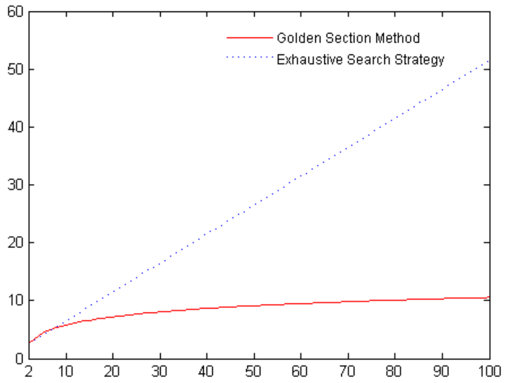

Fig. 1. The relationship between the length of search interval (abscissa) and the iteration time of the golden section algorithm or the exhaustive search algorithm (ordinate).

3.3.2. Parameter estimation

For each explored GMM structure in the line search process, the optimal parameters of the corresponding GMM are estimated according to Eq. (15). We apply the gradient ascent method to solve this maximization problem. Let $\boldsymbol{\Lambda}_t$ and α_t be the parameter set and the step size in the t -th iteration, respectively; $\nabla\widetilde{F}(\boldsymbol{\Lambda}_t, M)$ be the partial derivatives of $\widetilde{F}(\boldsymbol{\Lambda}_t, M)$ with respect to each parameter in $\boldsymbol{\Lambda}_t$. Then we have

$$\boldsymbol{\Lambda}_{t+1} = \boldsymbol{\Lambda}_t + \alpha_t \nabla\widetilde{F}(\boldsymbol{\Lambda}_t, M). \qquad (25)$$

$\boldsymbol{\Lambda}_t$ includes the classifier parameters and two soft targets, i.e., $\hat{H}$ and $\overline{H}$. So the two soft targets are adaptively adjusted according to Eq. (25) in each training iteration.

In order to reduce the overfitting problem and accelerate the speed of parameter estimation, we involve only training examples easily confused with each other in the parameter estimation procedure. It is realized by temporarily removing training examples, for which posterior pseudo-probabilities have distinctly exceeded the corresponding soft target. Let $\hat{\boldsymbol{S}}_t$ and $\overline{\boldsymbol{S}}_t$ be the set of positive and negative examples of the class C_i, which are involved in the t-th training iteration, respectively. Then the data removal schema can be expressed as

$$\begin{cases} \hat{\boldsymbol{S}}_t = \left\{ \hat{\boldsymbol{x}} \middle| f\left(p\left(\hat{\boldsymbol{x}} \middle| C_i\right)\right) \leq \hat{H} + \delta_t \wedge \hat{\boldsymbol{x}} \in \hat{\boldsymbol{S}}_{t-1} \right\} \\ \overline{\boldsymbol{S}}_t = \left\{ \overline{\boldsymbol{x}} \middle| f\left(p\left(\overline{\boldsymbol{x}} \middle| C_i\right)\right) \geq \overline{H} - \delta_t \wedge \overline{\boldsymbol{x}} \in \overline{\boldsymbol{S}}_{t-1} \right\} \end{cases}, \qquad (26)$$

where δ_t is a threshold value for determining whether the posterior pseudo-probability is distinctly exceed the soft target. Let $t_{\max}$ be the maximum times of training iterations; $\delta_{\max}$ and $\delta_{\min}$ be the maximum and minimum value of δ_t, respectively. Then δ_t in the t -th training iteration is

$$\delta_t = \delta_{\max} - t(\delta_{\max} - \delta_{\min}) / t_{\max} \ . \tag{27}$$

The removed examples in the t -th training iteration will be reinserted into the training set in the $(t+R)$-th training iteration. R is the required span of training iterations, in which the example is excluded from the training. Let R_0 be the minimum span, i be the times of an example being removed from the training set. Then R for this example is

$$R = iR_0 \ . \tag{28}$$

3.3.3. Algorithm steps

We summarize our GMM structure selection algorithm for each class in Table 1-2. The algorithm is composed of two-layer optimization procedure, the outer is for GMM structure selection and the inner is for parameter estimation of the GMM with fixed structure. The whole process of discriminative GMM structure selection is to perform this algorithm one by one for all the classes under consideration.

Table 1.

Soft-MMP parameter estimation algorithm.

```
Input: training data set, initial parameters of posterior pseudo-probability measure function, initial
    values of two soft targets, and the iteration number $t = 0$.

Optimization:
  Repeat
    Step 1    Compute the empirical loss of the current classifier on the training data set.
    Step 2    Remove the examples from the training set according to Eq. (26).
    Step 3    Compute the partial derivative of $\widetilde{F}(\boldsymbol{\Lambda}_t, M)$ with respect to each parameter using
              remaining examples.
    Step 4    Update the unknown parameters using Eq. (25).
    Step 5    Update $\delta$ using Eq. (27).
    Step 6    Reinsert the examples removed in the previous iterations based on Eq. (28).
    Step 7    $t = t + 1$.
  Until convergence or $t \geq t_{\max}$. Let $\varepsilon$ be an infinitesimal, then the convergence condition is
      $F(\boldsymbol{\Lambda}_t) - F(\boldsymbol{\Lambda}_{t+1}) \leq \varepsilon$.

Output: the estimated soft targets and parameters of posterior pseudo-probability measure function.
```

Table 2.

The GMM structure selection algorithm.

```
Input: $M_0$: triggering number of GMM components; $\eta$: search step size; $\nu > 1$:
    acceleration factor; $\lambda = 2$: terminal length of search interval.

Model Structure Selection:
  Step 1  Initialize the search interval $[a, b]$ by the advance-retreat method
    Step 1.1  Let the iteration number $t = 0$. Estimate parameters in the GMM by the
              Soft-MMP parameter estimation algorithm shown in Table 1 and compute
              $\zeta(M_t) = \log \int \widehat{F}(\boldsymbol{\Lambda}, M) d\boldsymbol{\Lambda}$ using Eq. (22).
    Step 1.2  Let $M_{t+1} = M_t + \eta$. Compute $\zeta(M_{t+1})$. If $\zeta(M_{t+1}) > \zeta(M_t)$, then go to
              Step 1.3, or else go to Step 1.4.
    Step 1.3  Let $\eta = \nu\eta$, $M = M_t$, $M_t = M_{t+1}$, $t = t + 1$, and go to Step 1.2.
    Step 1.4  If $t = 0$, then let $M_t = M_{t+1}$, $\eta = -\eta$, and go to Step 1.2. Otherwise, let
              $a = \min\{M, M_t\}$, $b = \max\{M, M_t\}$, and go to Step 2.
  Step 2  Reducing the search interval by the golden section method
    Step 2.1  Let $t = 1$, $a_t = a$, $b_t = b$.
    Step 2.2  Calculate two numbers in the interval $[a, b]$: $p_t = \lfloor a_t + 0.382(b_t - a_t) \rfloor$ and
              $q_t = \lceil a_t + 0.618(b_t - a_t) \rceil$.
    Step 2.3  Estimate parameters in the GMM with $p_t$ and $q_t$ components by the
              Soft-MMP parameter estimation algorithm shown in Table 1, respectively. Compute
              $\zeta(p_t)$ and $\zeta(q_t)$.
    Step 2.4  If $\zeta(p_t) < \zeta(q_t)$, go to Step 2.5, or else go to Step 2.6.
    Step 2.5  If $b_t - p_t \le \lambda$, terminate the search process and output the optimal GMM with
              $q_t$ components. Otherwise, let $a_{t+1} = p_t$, $b_{t+1} = b_t$, $p_{t+1} = q_t$,
              $q_{t+1} = \lceil a_{t+1} + 0.618(b_{t+1} - a_{t+1}) \rceil$, compute $\zeta(q_{t+1})$, and go to Step 2.7.
    Step 2.6  If $q_t - a_t \le \lambda$, terminate the search process and output the optimal GMM with
              $p_t$ components. Otherwise, let $a_{t+1} = a_t$, $b_{t+1} = q_t$, $q_{t+1} = p_t$,
              $p_{t+1} = \lfloor a_{t+1} + 0.382(b_{t+1} - a_{t+1}) \rfloor$, compute $\zeta(p_{t+1})$, then go to Step 2.7.
    Step 2.7  Let $t = t + 1$ and go to Step 2.2.

Output: the GMM with optimal structure and parameters
```

4. Experiments

We evaluate our method of GMM structure selection by applying it to handwritten digit recognition. The resultant digit classifier is tested on the well-known CENPARMI database [21] and MNIST database [22]. The CENPARMI database contains 4,000 training examples and 1,000 test examples, and the MNIST database contains 60,000 training examples and 10,000 test examples.

4.1. Digit Modeling and Learning

The 8-direction gradient features (e-grg) [24] are used to represent digits in the experiments for both CENPAMI and MNIST database. The original 200-D e-grg is compressed to 100-D by the Principal Component Analysis (PCA) technique to improve the computation efficiency. The orthogonal GMM technique [34] is further used to reduce the correlation among elements in the feature vectors. Then, the feature vectors for each digit class are assumed to be of the GMM with diagonal covariance matrix. As a result, the set of unknown parameters in the Soft-MMP learning of our digit classifier is

$$\boldsymbol{\Lambda} = \left\{\kappa, \beta, w_k, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}, \hat{H}, \overline{H}\right\}, k = 1, \cdots, M \ . \tag{29}$$

Some parameters in Eq. (29) must satisfy certain constraints, which are transformed to unconstrained domain for easier implementation. The constraints and transformation of parameters are listed in Table 3. A tiny variance value in covariance matrices of the GMM will lead to the computational instability of class-conditional probability density function. So we impose a positive minimum limit on variance value, which is denoted as τ in Table 3. Consequently, the transformed parameter set is

$$\widetilde{\boldsymbol{\Lambda}} = \{\widetilde{\kappa}, \widetilde{\beta}, \widetilde{w}_k, \boldsymbol{\mu_k}, \widetilde{\boldsymbol{\Sigma}}_k, h_1, h_2\}, k = 1, \cdots, M \ . \tag{30}$$

We use the discriminative model structure selection algorithm shown in Table 2 to determine the optimal M as well as other parameters in Eq. (30), and then transform them into the original ones.

Table 3

The constraints and transformation of parameters in the learning of digit classifiers.

Original parameters and constrains	Transformation of parameters
$0 < \hat{H} < 1;\ 0 < \overline{H} < 1$	$\hat{H} = \frac{1}{1+e^{-h_1}}; \overline{H} = \frac{1}{1+e^{-h_2}}$
$\kappa > 0;\ \beta > 0$	$\kappa = \exp(\tilde{\kappa}); \beta = \exp(\tilde{\beta})$
$\sigma_{kj} > \tau$	$\sigma_{kj} = \exp(\tilde{\sigma}_{kj}) + \tau$
$\sum w_k = 1$	$w_k = \frac{e^{\tilde{w}_k}}{\sum e^{\tilde{w}_k}}$

4.2. Experimental results

The parameters in our algorithm were set by experiments and listed in Table 4. These parameter values were used for both CENPARMI and MNIST test.

Table 4

Algorithm parameter setting.

Parameter	α_t	$t_{\max}$	λ	$\delta_{\max}$	$\delta_{\min}$	R_0	τ
Value	0.00001	30,000	0.1	0.5	0.01	50	0.001

The role and experiential setting method of each parameter in Table 4 are explained as follows. (1) α_t and $t_{\max}$ are the step size and the maximum iterative times for the gradient ascent optimization, respectively. They have important influence on training efficiency and effectiveness. Although some heuristic methods for setting α_t and $t_{\max}$ have been presented in literatures [35], the choice of them is mainly data dependent at present. (2) The parameter λ controls the tradeoff between two sub-objectives in the Soft-MMP learning criteria, i.e., the minimum empirical loss and the maximum difference between two soft targets. So λ should be adjusted to make the weighted values of two parts in Eq. (14) be close to each other. (3) The parameter τ is used to prevent the computational instability of class-conditional probability density function. It must be positive and be set as small as possible. (4) The other algorithm parameters, including $\delta_{\max}$, $\delta_{\min}$ and R_0, are used for data selection in the training process. The values of $\delta_{\max}$ and $\delta_{\min}$ should be large enough and small enough respectively, so that the training set can be exploited sufficiently at initial stages of training while more and more examples which have been learned well can be ignored at succedent training stages. As for R_0, the increase of its value will lead to more efficient algorithm but more risks of worsening training results. We determined the ideal R_0 by experiments. Although the proposed algorithm can work without the data selection procedure, the inclusion of it can improve training efficiency and effectiveness. In a previous work, we conducted the handwritten digit recognition experiments on the MNIST database by using the original Soft-MMP with or without data selection procedure. The experimental results show that the use of data selection can lead to better training efficiency and generalization ability which is measured as the ratio of the recognition rate on the test set to that on the training set. On a same computation platform, the training time was decreased from 7805 seconds to 3549 seconds, while the generalization ability was increased from 0.9959 to 0.9960.

4.2.1. Comparisons of model structure selection methods

Our discriminative model structure selection method is compared with manual setting method and three generative counterparts, including BIC [10], MDL [7] and AutoClass [23]. The GMM structures selected by automatic methods vary with different digit classes. We list the GMM structure selection results on the CENPAMI and MNIST database in Table 5-6, respectively. As shown in Table 5, the number of GMM components computed by our method averages around 4 for the CENPAMI database. So we perform three tests of manual setting on the CENPARMI database, in which 3 to 5 numbers of the GMM components are assigned to all the digit classes, respectively. For the MNIST database, 6-8 numbers of GMM components are considered in the manual setting tests because of the same reason.

Table 5

The GMM structure selected by BIC, MDL, AutoClass, and our method for each digit class on the CENPAMI database.

Classes Methods	0	1	2	3	4	5	6	7	8	9	Average
BIC	4	4	3	4	5	4	4	5	5	4	4.2
MDL	3	3	2	2	4	3	2	4	3	2	2.8
AutoClass	4	3	4	4	4	4	3	5	5	5	4.1
Our	4	3	3	5	3	3	5	4	6	5	4.1

Table 6

The GMM structure selected by BIC, MDL, AutoClass, and our method for each digit class on the MNIST database.

Methods \ Classes	0	1	2	3	4	5	6	7	8	9	Average
BIC	6	12	6	7	4	7	6	9	6	7	7.0
MDL	3	5	3	3	5	4	3	5	3	4	3.8
AutoClass	4	8	5	4	5	6	6	9	4	8	5.9
Our	5	10	4	4	6	5	4	7	9	10	6.4

Besides structure selection, the parameter estimation is another important problem for GMM modeling. In our method, the structure selection and the parameter estimation are completed simultaneously in a discriminative manner. However, the generative EM algorithm is used to estimate parameters in original BIC, MDL, and AutoClass. In order to fairly compare these three model structure selection methods and ours, the Soft-MMP discriminative learning algorithm is used to revise the parameters from original BIC, MDL and AutoClass, respectively.

Based on three model structures set manually and four model structures determined automatically, we get seven digit classifiers for the CENPARMI and MNIST database, respectively. Then the handwritten digit recognition is performed by using each of these seven classifiers on the training set and the test set, respectively. Table 7 shows the corresponding error rate on the training set (Train) and the test set (Test) for the CENPAMI database and Table 8 for the MNIST database. The generalization ability of each classifier is further measured as the ratio of the recognition rate on the test set to that on the training set. It is denoted as “Test/Train” in Table 7-8. The larger ratio value means the better generalization ability. In Table 7-8, we also list the reduction in the error rate on the test set (Reduction_test), which is brought by our method compared with other methods.

As shown in Table 7-8, our discriminative method of GMM structure selection achieves the better result than manual method as well as generative counterparts. Compared with BIC, MDL, and AutoClass, our method brings 27.78%, 40.91% and 38.10% reduction in the error rate on the CENPARMI test set, and 27.40%, 32.91% and 15.87% reduction in the error rate on the MNIST test set, respectively. Furthermore, our method improves the generalization ability from 0.9922 (BIC), 0.9912 (MDL) and 0.9915 (AutoClass) to 0.9940 (ours) on the CENPARMI database and from 0.9956 (BIC), 0.9958 (MDL) and 0.9960 (AutoClass) to 0.9966 (ours) on the MNIST database.

Table 7

Error rates from four automatic methods and the manual method of structure selection on the CENPAMI database, where 3-component to 5-component mean that 3 to 5 numbers of GMM components are manually assigned to all the digit classes, respectively.

Structure Selection Method	Train (%)	Test (%)	Reduction_test(%)	Test / Train
3-Components	0.300	1.35	51.85	0.9895
4-Components	0.225	1.15	43.48	0.9907
5-Components	0.200	1.15	43.48	0.9904
BIC	0.125	0.90	27.78	0.9922
MDL	0.225	1.10	40.91	0.9912
AutoClass	0.200	1.05	38.10	0.9915
Our	**0.050**	**0.65**	-	**0.9940**

Table 8

Error rates from four automatic methods and the manual method of structure selection on the MNIST database, where 6-component to 8-component mean that 6 to 8 numbers of GMM components are manually assigned to all the digit classes, respectively.

Structure Selection Method	Train (%)	Test (%)	Reduction_test(%)	Test / Train
6-Components	0.31	0.80	33.75	0.9951
7-Components	0.25	0.76	30.26	0.9949
8-Components	0.21	0.67	20.90	0.9954
BIC	0.29	0.73	27.40	0.9956
MDL	0.37	0.79	32.91	0.9958
AutoClass	0.23	0.63	15.87	0.9960
Our	**0.19**	**0.53**	-	**0.9966**

4.2.2. Comparisons to the state-of-the-art digit classifiers

The handwritten digit recognition rate achieved by our method is further compared with the state-of-the-art on the CENPAMI and MNIST database, respectively.

(1) CEMPAMI database

In the paper of Liu et al. [24-25], state-of-the-art techniques of handwritten digit recognition, including features and classifiers, are thoroughly investigated on the CENPARMI database. They reported their best error rate of 0.95% on the CENPARMI test set for e-grg features by using either SVM with RBF kernel or DLQDF. They also reported the overall best error rate of 0.85% on the CENPARMI test set, which comes from 8-direction deslant chaincode feature (des) instead of e-grg [25]. Our digit classifier achieves better result based on e-grg features, i.e. the error rate of 0.65% on the test set. Furthermore, we collect other up-to-date results on the CENPARMI database and compare them with ours in Table 9. It shows that the digit classifier trained by the proposed GMM structure selection method experimentally outperforms other counterparts.

Table 9
Error rates of various up-to-date digit classifiers on the CENPARMI database.

Classification Method	Feature	Test (%)
Modular Neural Network [26]	class dependent features	2.15
Local Learning Framework[27]	32-direction gradient features	1.90
Neural Network[28]	random features	1.70
Virtual SVM [29]	32-direction gradient features	1.30
SVC-rbf [24]	e-grg	0.95
SVC-rbf [25]	des	0.85
Our	**e-grg**	**0.65**

(2) MNIST database

Liu et al. [24-25] also compared state-of-the-art techniques of handwritten digit recognition on the MNIST database. They reported the best error rate of 0.42% on the test set for e-grg features by using SVM with RBF kernel. Our method achieves the comparable error rate of 0.53% by using the same features. Furthermore, we also collect other up-to-date results on the MNIST database and compare them with ours in Table 10. It shows that the performance of our method outperforms most of the state-of-art techniques and comparable to the currently best ones.

Table 10
Error rates of various up-to-date digit classifiers on the MNIST database.

Classification Method	Feature	Test (%)
Convolutional Net LeNet-1 [22]	Subsampling	1.7
Polynomial SVM [36]	32-direction gradient features	1.4
Boosted LeNet4 [37]	Subsampling	0.70
Large Convolutional Net [38]	Unsup features	0.62
SVM [39]	Vision-based feature	0.59
SVM [40]	Trianable feature	0.54
K-NN [41]	Shiftable edges	0.52
VSVM [29]	32 direction gradient features	0.44
SVC-rbf [24]	e-grg	0.42
Large Convolutional Net [30]	Trainable feature	0.39
Our	**e-grg**	**0.53**

5. Conclusions

In this paper, a discriminative structure selection method of Gaussian Mixture Model (GMM) has been proposed based on Bayesian structure selection framework and a discriminative learning criterion of Bayesian classifiers, called Soft target based Max-Min posterior Pseudo-probabilities (Soft-MMP). Our main contribution is to tailor and integrate the Soft-MMP objective function into Bayesian model structure selection framework with Laplace's approximation. The resultant model structure selection criterion is the maximum value of Laplace's approximation of integrated Soft-MMP function. By developing a line search algorithm to find out this maximum value, we simultaneously determine the structure of and the parameters in the optimal GMM.

The proposed GMM structure selection method was tested in handwritten digit recognition tasks. The experiments were conducted on the well-known CENPAMI and MNIST handwritten digit databases. Our method experimentally outperforms manual setting method and generative counterparts including Bayesian Information Criterion (BIC), Minimum Description Length (MDL) and AutoClass, both in recognition accuracy and generalization ability. Furthermore, to our best knowledge, the handwritten digit classifier trained by our method achieves the best recognition rate so far on the CENPARMI database and the comparable result to the currently best ones on the MNIST database.

The advantages of the proposed method are three-fold: (1) the discriminative criterion of structure selection is directly related to classification lose, so the method can work well on small data sets; (2)

by using the line search strategy instead of commonly used exhaustive search strategy, the method is suitable for large-scale structure selection problems; and (3) with the help of data selection schema, the computation is tractable even for training on large data sets. However, the proposed method gives more emphasis on the training data which are confused with each other, so its robustness to noise data seems inferior to that of generative counterparts.

In the future, we will evaluate the effectiveness of the proposed method in more applications, on more databases, and for other finite mixture models besides GMM.

Acknowledgements

This research was supported by National Natural Science Foundation of China (Grant No. 60973059, 90920009), Program for New Century Excellent Talents in University of China (Grant No. NCET-10-0044), and Excellent Young Scholars Research Fund of Beijing Institute of Technology (Grant No. 2008YS1203). The authors would like to thank Dr. Cheng-Lin Liu for providing the e-grg features of handwritten digits and the referees for their valuable comments and suggestions.

References

[1] M.A.T. Figueiredo, A.K. Jain, Unsupervised Learning of Finite Mixture Models, IEEE Trans. Pattern Analysis and Machine Intelligence 24(3) (2002) 381-396.

[2] G. J. McLachlan, On Bootstrapping the Likelihood Ratio Test Statistic for the Number of Components in a Normal Mixture, J. Royal Statistic Soc. C 36(1987) 318-224.

[3] P. Smyth, Model Selection for Probabilistic Clustering Using Cross-Validated Likelihood, Statistics and Computing 10(1) (2000) 63-72.

[4] H. Bensmai, G. Celeux, A. Raftery, and C. Robert, Inference in Model-Based Cluster Analysis, Statistics and Computing 7(2002) 1-10.

[5] H. Bozdogan, S.L. Sclove, Multi-sample cluster analysis using Akaike's information criterion, Annals of the Institute of Statistical Mathematics 36(1984) 163-180.

[6] C. Wallace, and D. Dowe, Minimum message length and Kolmogorov complexity, Computer Journal 42(4) (1999) 270-280.

[7] J.J. Rissanen, Information and Complexity in Statistical Modeling (Springer-Verlag, New York, 2007).

[8] C. E. Rasmussen, The Infinite Gaussian Mixture Model, Advances in Neural Information Processing Systems (2000) 554-560.

[9] H. Attias, Inferring Parameters and Structure of Latent Variable Models by Variational Bayes, in: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (1999).

[10] G. Schwarz, Estimating the dimension of a model, Annals of Statistics 6(1978) 461-464.

[11] D. J. C. MacKay, Choices of Basis for Laplace Approximation, Machine Learning 33(1) (1998) 77-86.

[12] A. Corduneanu, C.M. Bishop, Variational Bayesian Model Selection for Mixture Distributions, Artificial Intelligence and Statistics (T. Jaakkola & T.Richardson, Morgan Kaufmann, (2001).

[13] S.J. Roberts, D. Husmeier, I. Rezek, and W. Penny, Bayesian approaches to Gaussian modeling, IEEE Trans. Pattern Analysis and Machine Intelligence 20(1998) 1133-1142.

[14] B.H. Juang, W. Chou, and C.H. Lee, Minimum classification error rate methods for speech recognition, IEEE Trans. Speech Audio Processing 5(3) (1997) 257-265.

[15] R. Nopsuwanchai, A. Biem, and W. F. Clocksin, Maximization of Mutual Information for Offline Thai Handwriting Recognition, IEEE Trans. Pattern Analysis and Machine Intelligence 28(8) (2006) 1347-1351.

[16] D. Povey, and P.C. Woodland, Minimum phone error and I-smoothing for improved discriminative training, In: Proc. ICASSP, (2002) 105-108.

[17] X.B. Liu, Y.D. Jia, X.F. Chen, Y. Deng, and H. Fu, Image Classification Using the Max-Min Posterior Pseudo-Probabilities Method, Technical Report BIT-CS-20080001, Beijing Institute of Technology, http://mcislab.cs.bit.edu.cn/member/~xiabi/papers/2008_1.PDF, 2008.

[18] A. Klautau, N. Jevtic, and A. Orlitsky. Discriminative Gaussian Mixture Models: A comparison with kernel classifiers. Proceedings of the Twentieth International Conferenceo n MachineL earning (ICML-PO03)W, ashingtonD C, 2003.

[19] X.Y. Liu, and M. Gales, Automatic Model Complexity Control Using Marginalized Discriminative Growth Functions, IEEE Trans. Audio, Speech, and Language Processing 12(4) (2007) 1414-1424.

[20] X.F. Chen, X.B. Liu, and Y.D. Jia, A Soft Target Method of Learning Posterior Pseudo-probabilities based Classifiers with its Application to Handwritten Digit Recognition, In 2008 11th International Conference on Frontiers in Handwriting Recognition, 2008.

[21] C.Y. Suen, et al., Computer recognition of unconstrained handwritten numerals, in: Proc. IEEE 80(7) (1992) 1162-1180.

[22] Y. LeCun, et al., Comparison of Learning Algorithms for Handwritten Digit Recognition, Proceedings of The International conference on Artificial Neural Networks, Nanterre, France, (1995) 53-60.

[23] P. Cheeseman, and J. Stutz, Bayesian Classification (AutoClass): theory and results. Advances in knowledge discovery and data mining (AAAI Press, Menlo Park, CA. USA, 1996) 153-180.

[24] C.L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, Handwritten digit recognition: benchmarking of state-of-the-art techniques, Pattern Recognition 36(2003) 2271-2285.

[25] C.L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, Handwritten digit recognition: investigation of normalization and feature extraction techniques, Pattern Recognition 37(2004) 265-279.

[26] I.S. Oh, J.S. Lee, and C.Y. Suen, Analysis of class separation and combination of class-dependent features for handwriting recognition, IEEE Trans. Pattern Analysis and Machine Intelligence 12(10) (1999), 1089-1094.

[27] J.X. Dong, A. Krzyzak, C.Y. Suen, Local Learning framework for handwritten character recognition. Engineering Applications of Artificial Intelligence 15(2002) 151-159.

[28] P.D. Gader, M.A. Khabou, Automatic feature generation for handwritten digit recognition, IEEE Trans. Pattern Analysis and Machine Intelligence 18(12) (1996) 1256-1261.

[29] J.X. Dong, A. Krzyzk, C.Y. Suen, Fast SVM Training Algorithm with Decomposition on Very Large Datasets, IEEE Trans. Pattern Analysis and Machine Intelligence 27(4) (2005) 603-618 http://www.cenparmi.concordia.ca/~jdong/HeroSvm.html.

[30] Ranzato Marc Aurelio, Christopher Poultney, Sumit Chopra and Yann LeCun, Efficient Learning of Sparse Representations with an Energy-Based Model, Advances in Neural Information Processing Systems, MIT Press, 2006.

[31] G. Lugosi, and M. Pawlak, On the posterior-probability estimate of the error rate of nonparametric classification rules, IEEE Trans. Information Theory, 40(2) (1994) 475-481.

[32] G.J. McLachlan, and S.K. Ng, A comparison of some information criteria for the number of components in a mixture model, Technical Report, Department of Mathematics, University of Queensland, 2000.

[33] E. Polak, Optimization Algorithms and Consistent Approximations (Springer-Verlag, New York, USA, 1997).

[34] R. Zhang, X.Q. Ding, Offline Handwritten Numeral Recognition Using Orthogonal Gaussian Mixture Model, in: Proceedings 6th Int. Conference document Analysis and Recognition, Seattle, USA, (2001) 1126-1129.

[35] P. Baldi, Gradient descent learning algorithm overview: A general dynamical systems perspective, IEEE Transactions on Neural Networks, 6(1) (1995), 182-195.

[36] C.J.C. Burges and B. Scholkopf, Improving the accuracy and speed of support vector leaning machines, Advances in Neural information Processing Systems, MIT Press, (1997).

[37] Y. Lecun, L. BOttou, Y. Bengio and P. Haffner, Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11) (1998), 2278-2324.

[38] M.A. Ranzato, Fu-JIe Huang, Y.L. Boureau, and Yann Lecun, Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition, Proc. Computer Vision and Pattern Recognition Conf., (2007).

[39] L.N. Teow and K.F. Loe, Robust vision-based features and classification schemes for off-line handwritten digit recognition, Pattern Recognition,40(2002), 2355-2364.

[40] F.Lauer, C.Y. Suen and G. Bloch, A trainable feature extractor for handwritten digit recognition, Pattern Recognition, 40 (2007), 1816-1824.

[41] Daniel Keysers, Thomas Deselaers, Christian Gollan, and Hermann Ney, Deformation Models for Image Recognition, IEEE Trans. Pattern Analysis and Machine Intelligence, 29(8) (2007), 1422-1435.

Fig. 2

Fig. 3

Fig. 5